TRPO(Trust Region Policy Optimization) (1)

https://arxiv.org/pdf/1502.05477

▼ Introduction

정책 최적화를 위한 알고리즘은 3가지 카테고리로 분류된다.

1. policy iteration method
2. policy gradient method
3. derivative-free optimization

연속적 기울기 기반 최적화는 지도학습에서 다수의 파라미터를 갖는 지도학습 문제에 대해 성공적으로 함수를 근사시켰다. 이를 강화학습에도 적용하여 고차원적인 정책학습이 가능하게 하려한다.

▼ Preliminaris

S : state

A : action

P : S에서 A를 취해 다음상태 S′로 갈 확률

r : reward

ρ_0 : 초기에 어느 상태에서 시작할지에 대한 확률분포

γ : 감가율

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right], \text{ where}$$
$$s_0 \sim \rho_0(s_0),\ a_t \sim \pi(a_t|s_t),\ s_{t+1} \sim P(s_{t+1}|s_t, a_t).$$

$\eta(\pi)$는 시작시점부터 종단시점까지의 리워드에 감가율을 곱해 더해준 것으로 해당 정책의 성능지표이다.

$$
\begin{aligned}
Q_\pi(s_t, a_t) &= \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right], \\
V_\pi(s_t) &= \mathbb{E}_{a_t, s_{t+1}, \ldots}\left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})\right], \\
A_\pi(s, a) &= Q_\pi(s, a) - V_\pi(s), \text{ where} \\
& a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t) \text{ for } t \geq 0.
\end{aligned}
$$

$Q_\pi(s_t, a_t)$, $V_\pi(s_t)$는 해당 시점(t) 이후 취할 수 있는 보상의 가중합이다.

$A_\pi(s, a)$는 미래 액션의 가치에서 현 상태의 보상 추정값을 빼어 줌으로써 해당 액션의 가치를 추정할 수 있다.

$$
\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \cdots \sim \tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right]
$$

$\eta(\tilde{\pi})$는 업데이트 된 정책 $\tilde{\pi}$를 이전 정책 π를 이용해 가치를 얻어낸다. 샘플링은 업데이트 된 정책에서 하고,

가치는 이전 정책에 대해서 매기게 되면 이것이 곧 $\eta(\tilde{\pi})$가 되었다.

$$
\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \ldots,
$$

ρ_0는 시작지점을 어디로 잡을지에 대한 확률분포였다.

$\rho_\pi(s)$는 정책π를 따를 때 s라는 state에 몇번째 차례에 방문할지를 나타내는 확률분포이다.

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s)\gamma^t A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a)$$

$$= \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a). \qquad (2)$$

앞서 나온 $\eta(\tilde{\pi})$ 수식에서 기대값(E)를 푼 수식이다. state에서 a를 골랐을 때 A의 기대값이기에 이를 풀어주면 각 state에 도달할 확률과 해당 state에서 정책에 따라 action을 고를 확률이 또 곱해져야한다. 이를 수식으로 보이면 위 식과 같이 된다.

강화학습의 목표는 성능이 향상되는 수식에 따라 정책을 업데이트함으로써 최적의 정책(π^*)을 뽑아내는 것이다. 하지만 마지막 수식을 보게되면 $\rho_{\tilde{\pi}}(s)$ > 0 이고, $\tilde{\pi}(a|s)$도 확률이기에 0보다 크다.

결론적으로 $A_\pi(s,a)$가 ≥ 0 이면 어느 한 쌍의 action-state쌍만 향상되더라도 해당 정책은 성능 향상이 보장된다.(쇠퇴하지 않는다.)

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_{s} \rho_\pi(s) \sum_{a} \tilde{\pi}(a|s) A_\pi(s,a)$$

$$1) \sum_{s} \rho_{\tilde{\pi}(s)} \sum \tilde{\pi}(a|s) A_\pi(s,a)$$

$$2) \sum_{s} \rho_{\pi}(s) \sum \tilde{\pi}(a|s) A_\pi(s,a)$$

1) 수식과 2)수식의 차이는 state에 방문할 확률분포가 현재 정책을 따르는지, 미래 정책(업데이트 된 정책)을 따르는지이다.

논문에서는 근사과정에서 평가오차와 근사오차로 인해 $A_\pi(s,a)$ 텀이 음수가 될 수 있다고 말을 한다.
gym 환경의 frozen lake 게임을 생각해보자. 에이전트에게 시킨 action과 다른 action이

일정 확률로 발생한다. 이렇듯 에이전트가 state와 action이 정해졌을 때 s′에 갈 확률 P가 정의되는 것은 이러한 오류때문에 그렇다. 각 정책에 이런 오류도 다 포함되어 있을텐데 $\rho_{\tilde{\pi}}$는 $P_{\tilde{\pi}}(s)$의 확률을 따른다. 하지만 $A_{\pi}(s,a)$는 $P_{\pi}(s)$를 따르고 있기에 우리가 명령한 것과 다른 움직임을 보이면 음의 값이 뜰 수 있다.

그렇기에 정책을 π로 바꿔준, 2번째 수식이 등장하였다.

이런 차이때문에 두 공식을 $\eta(\tilde{\pi})$, $L_{\pi}(\tilde{\pi})$라고 표기한다.

$$L_{\pi_{\theta_0}}(\pi_{\theta_0}) = \eta(\pi_{\theta_0}),$$
$$\nabla_{\theta} L_{\pi_{\theta_0}}(\pi_{\theta})\big|_{\theta=\theta_0} = \nabla_{\theta}\eta(\pi_{\theta})\big|_{\theta=\theta_0}.$$

파라미터화 시킨 초기의 정책 π하 L과 η는 동일하다. 또한 파라미터화가 되었기에 미분이 가능해진다.
그래서 위 수식은 성립한다.

작은 스텝사이즈로 최적화를 하게 되면 정책이 향상됨을 앞선 과정에 따라 보일 수 있다. 하지만 스텝사이즈가 커진다면 어떻게 될까?

$$\pi' = \arg\max_{\pi'} L_{\pi_{\text{old}}}(\pi')$$

$$\pi_{\text{new}}(a|s) = (1-\alpha)\pi_{\text{old}}(a|s) + \alpha\pi'(a|s)$$

큰 스텝 사이즈의 최적화를 이렇게 일정 비율(α)만큼만 업데이트하게 반영시켜 작은 스텝사이즈로 업데이트 하는 것과 유사한 효과를 내게 한 수식이다.

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{2\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$\text{where } \epsilon = \max_s \left| \mathbb{E}_{a\sim\pi'(a|s)} \left[A_\pi(s,a) \right] \right|$$

다음과 같은 lower bound를 갖는다고 한다.

▼ Monotonic Improvement Guarantee for General Stochastic Policies

$$D_{TV}(p \parallel q) = \frac{1}{2}\sum_i |p_i - q_i|$$

$$D_{\text{TV}}^{\max}(\pi, \tilde{\pi}) = \max_s D_{TV}(\pi(\cdot|s) \parallel \tilde{\pi}(\cdot|s))$$

두 정책에 의해 샘플링된 확률중 가장 큰 차이 → 정책간의 가장 큰 차이

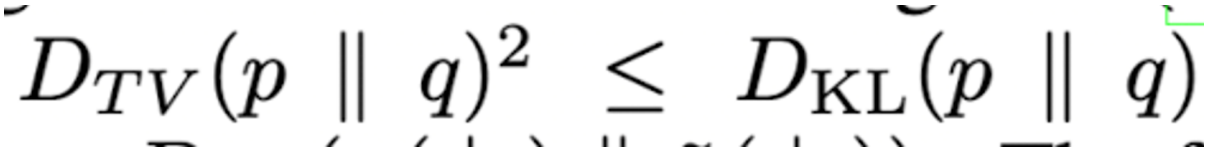
두 정책간에 얼마만큼 차이가 있는지(성능이 향상되었는지) 확인할 수 있는 지표로 total variation divergence를 보여주었다.

$$\alpha = D_{\text{TV}}^{\max}(\pi_{\text{old}}, \pi_{\text{new}}).$$

$$\eta(\pi_{\text{new}}) \geq L_{\pi_{\text{old}}}(\pi_{\text{new}}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha^2$$

$$where\ \epsilon = \max_{s,a} |A_\pi(s,a)|$$

이를 이용하면 앞서 나온 lower bound를 다음과 같은 식으로 변경할 수 있다.



$$D_{TV}(p \parallel q)^2 \leq D_{\mathrm{KL}}(p \parallel q)$$

KL divergence 와 TV divergence는 다음과 같은 공식이 성립한다.

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - CD_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi}),$$
$$\text{where } C = \frac{4\epsilon\gamma}{(1-\gamma)^2}.$$

앞선 부등식을 적용시키고, 상수를 C로 묶어주게 되면 결론적으로 다음 식이 나온다.

$$L_{\pi_i}(\pi) - CD_{\mathrm{KL}}^{\max}(\pi_i, \pi).$$

$M(\pi_i)$는 다음과 같이 정이되었다.

정책이 향상됨을 가정하게 되면 아래 부등식이 성립하게 된다.

$$\eta(\pi_{i+1}) \geq M_i(\pi_{i+1}) \text{ by Equation (9)}$$
$$\eta(\pi_i) = M_i(\pi_i), \text{ therefore,}$$
$$\eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M(\pi_i).$$

그러면 $M(\pi_i)$(panelty term을 포함하는 성능지표)가 극대화되면 η는 감소하지 않는다.

결론적으로 다음 알고리즘이 정의된다.

Algorithm 1 Policy iteration algorithm guaranteeing non-decreasing expected return η

Initialize π_0.

for $i = 0, 1, 2, \ldots$ until convergence **do**

Compute all advantage values $A_{\pi_i}(s, a)$.

Solve the constrained optimization problem

$$\pi_{i+1} = \arg\max_{\pi} \left[L_{\pi_i}(\pi) - C D_{\mathrm{KL}}^{\max}(\pi_i, \pi) \right]$$

where $C = 4\epsilon\gamma/(1-\gamma)^2$

and $L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$

end for

▼ Optimization of Parameterized Policies

본 섹션에서는 TRPO (Trust Reston Policy Optimization)을 소개한다.

큰 규모의 업데이트에도 로버스트한 성질을 띄게 하려고 패널티항 보다는 KL divergence 제약조건을 넣었다.

최적화문제는 다음과 같이 정의된다.

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta) \\ &\text{subject to } D_{\mathrm{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta. \end{aligned}$$

제약조건으로 표현한 최적화문제

$$\underset{\theta}{\text{maximize}} \left[L_{\theta_{\text{old}}}(\theta) - C D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \right].$$

panelty term을 이용한 최적화문제 정의

$$\overline{D}_{\text{KL}}^{\rho}(\theta_1, \theta_2) := \mathbb{E}_{s \sim \rho} \left[D_{\text{KL}}(\pi_{\theta_1}(\cdot|s) \parallel \pi_{\theta_2}(\cdot|s)) \right].$$

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta)$$
$$\text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.$$

$\rho_{\theta_{old}}$에서 각 state에 도달할 확률 * (각 정책 하에서 해당 state에서 가능한 action을 뽑아낼 확률값의 차) ≤ δ일 때 L을 최대로하는 θ를 구하는게 목표

수식을 풀어쓰면 다음과 같다.

$$\underset{\theta}{\text{maximize}} \sum_{s} \rho_{\theta_{\text{old}}}(s) \sum_{a} \pi_{\theta}(a|s) A_{\theta_{\text{old}}}(s, a)$$
$$\text{subject to } \overline{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}}, \theta) \leq \delta.$$

해당 수식에 따라 업데이트 된 정책이 θ인데, 업데이트를 하기 전에 θ에서 액션을 샘플링을 하는 것은 불가능하다. 따라서 해당 수식을 업데이트 전 정책에서 업데이트할 수 있게 바꿔야하고, 여기서 Importance sampling이 사용된다.

$$\sum_{a} \pi_{\theta}(a|s_n) A_{\theta_{\text{old}}}(s_n, a) = \mathbb{E}_{a \sim q} \left[\frac{\pi_{\theta}(a|s_n)}{q(a|s_n)} A_{\theta_{\text{old}}}(s_n, a) \right]$$

$$\underset{\theta}{\text{maximize}}\, \mathbb{E}_{s \sim \rho_{\theta_{\text{old}}}, a \sim q} \left[\frac{\pi_{\theta}(a|s)}{q(a|s)} Q_{\theta_{\text{old}}}(s, a) \right] \qquad (14)$$
$$\text{subject to } \mathbb{E}_{s \sim \rho_{\theta_{\text{old}}}} \left[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|s) \parallel \pi_{\theta}(\cdot|s)) \right] \leq \delta.$$

위 수식이 최종적인 TRPO 최적화 수식이 되겠다.

$A_{\theta old} = Q_{\theta old}(s, a) - V_{\theta old}(s)$이고, 이때 $V_{\theta old}$는 action의 영향을 받지않기에 기대값 내에서 상수로 취급된다. 상수텀은 함수의 수렴에 영향을 주지 않기에 제거해주면 (14)가 된다. (상수텀은 유무 관계없이 동일 함수로 수렴하게 된다.)